LLNL-TR-411194

LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

Factoring Algebraic Error for Relative Pose Estimation

Peter Lindstrom, Mark Duchaineau

March 11, 2009

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Factoring Algebraic Error for Relative Pose Estimation

Peter Lindstrom
pl@llnl.gov

Mark Duchaineau
duchaine@llnl.gov

Lawrence Livermore National Laboratory

Abstract

We address the problem of estimating the relative pose, i.e. *translation and rotation, of two calibrated cameras from image point correspondences. Our approach is to factor the nonlinear algebraic pose error functional into translational and rotational components, and to optimize translation and rotation independently. This factorization admits subproblems that can be solved using direct methods with practical guarantees on global optimality. That is, for a given translation, the corresponding optimal rotation can directly be determined, and vice versa. We show that these subproblems are equivalent to computing the least eigenvector of second- and fourth-order symmetric tensors. When neither translation or rotation is known, alternating translation and rotation optimization leads to a simple, efficient, and robust algorithm for pose estimation that improves on the well-known 5- and 8-point methods.*

1. Introduction

The problem of estimating the relative pose between two calibrated cameras has received considerable attention in the computer vision literature over the past several decades. This problem can be formulated as follows: Given two cameras with known intrinsic parameters and a set of point correspondences between matching pixels in the images captured by the cameras, determine the extrinsic parameters of the cameras, that is, their relative positions and orientations. As the point correspondences are often perturbed by noise, *e.g.* due to numerical or measurement errors, the pose estimation problem usually becomes one of determining the extrinsic parameters that best explain the observed points, as measured by an error functional dependent on the pose parameters.

In this paper we consider the minimization of what has become known as the *algebraic error* [5], as is also the goal of a number of prior methods [4, 5, 7, 14]. Though more elaborate functionals exist [6], a pose that minimizes algebraic error is usually a good approximation that may be used to seed secondary nonlinear optimization, and hence developing efficient approaches to algebraic error minimization has merit. The algebraic error can be written mathematically as follows. Let $x = [x_1 \;\; x_2 \;\; 1]^\mathsf{T}$ and $x' = [x'_1 \;\; x'_2 \;\; 1]^\mathsf{T}$ be corresponding points on the image plane in the respective coordinate frames of the two cameras. These points are related by the epipolar geometry constraint [11]

$$x^\mathsf{T} E x' = 0, \tag{1}$$

where

$$E = [t]_\times R \tag{2}$$

is a 3×3 rank-2 matrix called the *essential matrix*. Here t is a unit vector that represents the relative position, or translation, between the two cameras, and R is an orthogonal rotation matrix that represents relative camera orientation, both expressed in the coordinate frame of point x. Note that absent additional constraints the translation can be determined only up to scale, and as is common we choose $\|t\| = 1$. The skew-symmetric matrix $[t]_\times$ is defined as

$$[t]_\times = \begin{bmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{bmatrix}, \tag{3}$$

such that $[t]_\times x = t \times x$ for any x. In practice, Equation 1 is seldom satisfied exactly due to noise in x and x', and consequently we define the algebraic error functional $f(R, t)$ as a sum of square residuals

$$f(R, t) = f(E) = \sum_i (x_i^\mathsf{T} E x'_i)^2 \tag{4}$$

over all point pairs $\langle x_i, x'_i \rangle$. Minimizing f amounts to determining the nine elements of E. Because t is a unit-norm 3-vector, and since any rotation R can be expressed by as few as three parameters, E has only five degrees of freedom and its elements are related by a set of nonlinear constraints [13]. As a result of this, minimizing f is a nontrivial optimization problem.

Prior approaches to minimizing f can roughly be divided into two camps: direct [4, 10, 13–15, 21] and nonlinear methods [7, 12, 22]. Methods in the latter camp, such

Prepared by LLNL under Contract DE-AC52-07NA27344.

as *sparse bundle adjustment*, employ standard nonlinear iterative techniques [16, 17], possibly by using first and even second derivative information on f, and are subject to the usual pitfalls of high-dimensional nonlinear optimization, such as getting trapped in a local minimum. Direct methods [4, 14, 15], on the other hand, use linear algebra techniques such as SVD to obtain an approximate solution that typically violates the rank-2 constraint on E. A subsequent "projection" step is thus needed that in general leads to a suboptimal solution. The five-point method [10, 13, 21] avoids projection by explicitly incorporating the essential constraints, but restricts the optimization to a 4D subspace that, in the overconstrained case of more than five points, generally does not contain the global optimum.

In this paper we factor the optimization problem into separate eigenvalue subproblems that can be solved by simple (multi-)linear algebra techniques. For a fixed orientation R, this decomposition allows the optimal translation $t^* = \operatorname{argmin}_t f(R,t)$ for R to be computed; a known result due to Spetsakis and Aloimonos [20]. Our main contribution is to develop a solution to the reverse problem: for a fixed translation t, determine the optimal orientation $R^* = \operatorname{argmin}_R f(R,t)$. Our solution is of practical value in applications in which the relative camera positions are known, as incorporating such constraints allows determining a more accurate pose. We demonstrate the improved accuracy of our technique over Hartley's direct method [5] for known epipoles, as well as the unconstrained 5- and 8-point methods. A secondary contribution of this paper is a new iterative pose estimator that combines our globally optimal techniques to optimize translation and orientation alternately in 2D and 3D subspaces of 5D pose space. This method is stable and robust, requires neither derivatives nor heuristic step sizes, and provably reduces the algebraic error monotonically in each step. We show that using reasonable initial estimates, *e.g.* from existing direct methods, this process in practice converges quickly to a global minimum.

2. Preliminaries

Before describing our approach, we introduce some notation. Vectors and points are denoted by lowercase letters, *e.g.* x, matrices by uppercase, *e.g.* A, and higher-order tensors by caligraphic letters, *e.g.* $\mathcal{A}$. a_{ij} denotes the element in the i^{th} row, j^{th} column of matrix A, and tensor elements appear as Greek symbols, *e.g.* α_{ijkl}.

We write the *Kronecker product* of an $m \times n$ matrix A and a $p \times q$ matrix B as:

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix}. \tag{5}$$

The *vectorization* operator stacks the columns of an $m \times n$ matrix to form a vector of length mn, *e.g.*

$$\operatorname{vec} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{21} & a_{12} & a_{22} \end{bmatrix}^\mathsf{T}. \tag{6}$$

This operator satisfies the following identities:

$$\begin{aligned} \operatorname{vec}(ABC) &= (C^\mathsf{T} \otimes A)\operatorname{vec}(B) \\ \operatorname{vec}(AB) &= (I \otimes A)\operatorname{vec}(B) = (B^\mathsf{T} \otimes I)\operatorname{vec}(A) \\ \operatorname{tr}(A^\mathsf{T} B) &= \operatorname{vec}(A)^\mathsf{T} \operatorname{vec}(B), \end{aligned} \tag{7}$$

where I denotes the identity matrix. We will use unvec to denote the inverse of vec. For a full treatment of these operators, we refer the interested reader to [23].

3. Translation optimization

Let us first consider the problem of determining the translation t^* that minimizes f for a given rotation R, *i.e.*,

$$t^*(R) = \operatorname*{argmin}_{\|t\|=1} f(R,t) \tag{8}$$

We begin by rearranging $f(R,t) = f(E)$ as follows:

$$\begin{aligned} f(E) &= \sum_i (x_i^\mathsf{T} E x_i')^2 = \sum_i \operatorname{tr}(x_i'^\mathsf{T} E^\mathsf{T} x_i x_i^\mathsf{T} E x_i') \\ &= \sum_i \operatorname{tr}(x_i' x_i'^\mathsf{T} E^\mathsf{T} x_i x_i^\mathsf{T} E) \\ &= \sum_i \operatorname{vec}(x_i x_i^\mathsf{T} E x_i' x_i'^\mathsf{T})^\mathsf{T} \operatorname{vec}(E) \\ &= \sum_i \left((x_i' x_i'^\mathsf{T} \otimes x_i x_i^\mathsf{T})\operatorname{vec}(E)\right)^\mathsf{T} \operatorname{vec}(E) \\ &= \operatorname{vec}(E)^\mathsf{T} \left(\sum_i x_i' x_i'^\mathsf{T} \otimes x_i x_i^\mathsf{T}\right) \operatorname{vec}(E) \\ &= \operatorname{vec}(E)^\mathsf{T} X \operatorname{vec}(E), \end{aligned} \tag{9}$$

where we have made use of the identities in Equation 7 and the fact that the trace of a matrix product is invariant under cyclic permutations. Here X is a 9×9 symmetric positive semi-definite *coefficient matrix* that captures all of the point correspondences $\langle x_i, x_i' \rangle$, and allows evaluating f in constant time regardless of the number of points. Clearly the minimum of $f(E)$ is attained by letting $\operatorname{vec}(E)$ be the eigenvector corresponding to the smallest eigenvalue of X—this is the basis for the eight-point algorithm [4]. Such an eigenvector, however, cannot in general be decomposed in the form of Equation 2. Instead, we will explicitly decompose E into a valid translation and rotation.

Combining Equation 2 and Equation 7, we have

$$\operatorname{vec}(E) = (R^\mathsf{T} \otimes I)\operatorname{vec}([t]_\times) = (R^\mathsf{T} \otimes I)V_t t, \tag{10}$$

where V_t is given in Table 1. Plugging this back into Equation 9, we obtain

$$f(R,t) = t^\mathsf{T} V_t^\mathsf{T} (R \otimes I) X (R^\mathsf{T} \otimes I) V_t t = t^\mathsf{T} A_{(R)} t. \tag{11}$$

$$V_t = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad V_R = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Table 1. Vectorization matrices V_t and V_R.

We see that $A_{(R)}$ is a symmetric positive semi-definite 3×3 matrix that depends only on R (*i.e.* not on t), and hence the minimum of f is given by the smallest eigenvalue of $A_{(R)}$. The optimal translation t^* is therefore the corresponding (unit) eigenvector. Our implementation uses a small number of Jacobi rotations to find the full eigendecomposition of $A_{(R)}$, though other standard techniques such as inverse iteration could also be used [3]. Note that t^* is determined only up to sign as both $\pm t^*$ are eigenvectors. To find the correct translation, a simple cheirality test is used [6, 13].

Although quite simple, the above factorization of f appears to have received relatively little attention in the vision community, and we are aware of only one prior use of this formula. Spetsakis and Aloimonos [20] use the above result to design a Newton-like iteration scheme for the outer optimization problem of determining the rotation R. In the following section we show how a generalization of the approach here can be used to also formulate the optimal rotation as an eigenvalue problem.

4. Rotation optimization

We have seen above that one can factor the algebraic error minimization into two nested problems: $\min_R \min_t f(R,t)$. It is natural to ask if swapping the inner and outer problems leads to a similar formulation, *i.e.* given some translation t, can one efficiently compute $R^*(t) = \operatorname{argmin}_R f(R,t)$? The answer is yes, and the problem again reduces to an eigenvalue problem, although involving higher-order tensors.

We begin by applying the second identity of Equation 7 in the vectorization of E:

$$\operatorname{vec}(E) = \operatorname{vec}([t]_\times R) = (I \otimes [t]_\times) \operatorname{vec}(R). \tag{12}$$

Unlike $\operatorname{vec}([t]_\times)$, $\operatorname{vec}(R)$ cannot be written as a linear expression of independent parameters, *i.e.* R has only three degrees of freedom but nine interdependent elements. We will address this problem in part by using a quaternion parameterization of R. Let $q = [w \;\; x \;\; y \;\; z]^\mathsf{T}$ be a unit quaternion corresponding to R, such that

$$R = \begin{bmatrix} w^2+x^2-y^2-z^2 & 2xy-2wz & 2wy+2xz \\ 2xy+2wz & w^2-x^2+y^2-z^2 & 2yz-2wx \\ 2xz-2wy & 2wx+2yz & w^2-x^2-y^2+z^2 \end{bmatrix}. \tag{13}$$

Note that each element of R is a homogeneous quadratic polynomial in q. Hence we can write $\operatorname{vec}(R)$ as

$$\operatorname{vec}(R) = V_R(q \otimes q), \tag{14}$$

where V_R is the 9×16 constant matrix given in Table 1. This leads to the functional

$$\begin{aligned} f(q,t) &= (q \otimes q)^\mathsf{T} V_R^\mathsf{T} (I \otimes [t]_\times^\mathsf{T}) X (I \otimes [t]_\times) V_R (q \otimes q) \\ &= (q \otimes q)^\mathsf{T} A_{(t)} (q \otimes q), \end{aligned} \tag{15}$$

where $A_{(t)}$ is a 16×16 real, symmetric positive semi-definite matrix of rank at most six. We may rewrite Equation 15 in tensor form as

$$f(q,t) = \mathcal{A}_{(t)} q^4 = \sum_i \sum_j \sum_k \sum_l \alpha_{ijkl} q_i q_j q_k q_l, \tag{16}$$

where $\mathcal{A}_{(t)}$ is a fourth-order, four-dimensional tensor, and where the elements of $\mathcal{A}_{(t)}$ and $A_{(t)}$ are related by

$$\alpha_{ijkl} = a_{4(i-1)+j,4(k-1)+l}. \tag{17}$$

$f(q,t)$ is a quartic homogeneous polynomial, which in general has nontrivial minima. However, we will exploit the particular structure of this problem to arrive at a simple solution.

A tensor $\mathcal{A}_{(t)}$ is said to be *symmetric*[1] if α_{ijkl} is invariant under all permutations of the indices $ijkl$. Symmetry for tensors has the same implications on existence of real eigenvalues as it does for matrices. In general $\mathcal{A}_{(t)}$ is not symmetric, but we can compute a symmetric tensor $\bar{\mathcal{A}}_{(t)}$ by replacing each α_{ijkl} in $\mathcal{A}_{(t)}$ with $\frac{1}{4!}(\alpha_{ijkl} + \alpha_{ijlk} + \cdots + \alpha_{lkji})$ (see also [2, 18]). This operation does not affect the function value, *i.e.* $f(q,t) = \mathcal{A}_{(t)} q^4 = \bar{\mathcal{A}}_{(t)} q^4$ for all q. (This is analogous to the identity $x^\mathsf{T} A x = \frac{1}{2} x^\mathsf{T}(A + A^\mathsf{T})x$ for matrices.) Thus, w.l.o.g., we will assume from here on that all tensors are symmetric or can be symmetrized. We note that this symmetrization generally makes the unfolded matrix $A_{(t)}$ indefinite on $\mathbb{R}^{16}$ (*i.e.* $A_{(t)}$ may have negative eigenvalues), but of course $A_{(t)}$ is still positive semi-definite on the space of Kronecker squares $q \otimes q$.

[1] Also called '*supersymmetric*' by some authors (*c.f.* [8, 18]).

The quartic optimization problem

$$\min_{\|q\|=1} f(q,t) = \mathcal{A}_{(t)} q^4 \tag{18}$$

is a fourth-order eigenvalue problem that has been studied recently by Qi and others; see [19] and references therein. For real symmetric tensors of even order it is known that the minimum is given by the smallest *Z-eigenvalue* λ_{min}, and that such eigenvalues and corresponding eigenvectors exist [18]. In fact, the stationary points of $f(q,t)$ for fixed t are exactly the set of *Z-eigenvectors* of $\mathcal{A}_{(t)}$. In spite of the similarities with the second-order case (*i.e.* symmetric matrices), higher-order tensors do not share all the nice properties of matrix eigendecompositions. For example, the Z-eigenvectors of higher-order tensors are not necessarily orthogonal; there is no known computable canonical representation analogous to diagonalization that reveals the eigenvalues; and the number of eigenvalues for an m^{th}-order, n-dimensional tensor may be as many as $n(m-1)^{n-1}$ (in our case, $m = n = 4$), and may thus exceed n when $n, m \geq 3$ [18]. (In practice, we have found the number of eigenvalues to be in the range 12–24.) Nevertheless, it is possible to compute λ_{min} and the minimizer q^* with extremely high likelihood using the following procedure.

4.1. Higher-order eigenvector computation

We seek to find a q such that $\mathcal{A}_{(t)} q^3 = \lambda q$. As in the case of matrix eigendecompositions, the application $q \leftarrow \mathcal{A}_{(t)} q^3$ tends to steer q in the direction of the eigenvector of $\mathcal{A}_{(t)}$ with largest eigenvalue, which is the basis for the *power method* [3] and its generalization (HOPM) to higher-order tensors [9]. We are interested, however, in the eigenvector corresponding to the smallest eigenvalue, which can be computed using HOPM by constructing an auxiliary tensor $\rho\mathcal{I} - \mathcal{A}_{(t)}$, whose eigenvectors coincide with those of $\mathcal{A}_{(t)}$, but which in effect has the order of eigenvalues reversed. That is, $(\rho\mathcal{I} - \mathcal{A}_{(t)})q^3 = \rho q - \mathcal{A}_{(t)} q^3 = (\rho - \lambda) q$. Here ρ is a bound on the largest eigenvalue of $\mathcal{A}_{(t)}$, which is trivially obtained as the largest eigenvalue of the matrix $A_{(t)}$. The tensor $\mathcal{I}$ corresponds to the matrix unfolding $I_4 \otimes I_4 = I_{16}$, with elements related by Equation 17 (see also [2]).

Finally, although HOPM is guaranteed to converge [8], fast convergence requires a good initial guess for the eigenvector sought. Kofidis and Regalia [8] propose a solution to a similar problem by considering the eigendecompositions of the 4×4 matrices U_i formed by folding the eigenvectors u_i of $A_{(t)}$, *i.e.* $U_i = \text{unvec}(u_i)$. With some abuse of language, we will refer to these matrices U_i as the "eigenmatrices" of $A_{(t)}$. Kofidis and Regalia show that ten of the U_i are symmetric; the remaining six are skew symmetric and lie in the null space of $A_{(t)}$ (and are thus not of interest to us). Furthermore, the structure of our problem is such that four of the ten symmetric eigenmatrices are orthogonal (modulo scale), and three of these have zero trace.

```
1.  Compute $\mathcal{A}_{(t)}$, $A_{(t)}$ from $t$                (Equation 15)
2.  $\mathcal{A}_{(t)} = \bar{\mathcal{A}}_{(t)}$; $A_{(t)} = \bar{A}_{(t)}$      symmetrize
3.  $U \,\text{diag}(\mu_1, \ldots, \mu_{16}) U^\mathsf{T} = A_{(t)}$      eigendecomp.
4.  $\rho = \mu_1$                       upper bound on $\lambda_1(\mathcal{A}_{(t)})$
5.  $U_{16} = \text{unvec}(u_{16})$              fold least eigenvector
6.  $V \,\text{diag}(\nu_1, \ldots, \nu_4) V^\mathsf{T} = U_{16}$       eigendecomp.
7.  for $i = 1, 2$                       use two seed points
8.      $q_0 = v_i$                  initialize with eigenvector of $V$
9.      for $k = 1, 2, \ldots$                 power iteration loop
10.         $q_k = \rho q_{k-1} - \mathcal{A}_{(t)} q_{k-1}^3$
11.         $q_k = q_k / \|q_k\|$
12.     $f = \mathcal{A}_{(t)} q^4$
13.     if $f < f^*$ then                   keep best solution
14.         $f^* = f$
15.         $q^* = q$
```

Table 2. Algorithm for computing the optimal unit quaternion $q^*(t) = \text{argmin}_q f(q,t)$ for a given translation t.

Kofidis and Regalia choose as initial guess for the *maximizer* of f the dominant eigenvector of the dominant eigenmatrix U_1 of $A_{(t)}$. The analogous solution to our problem is to choose the dominant eigenvector of U_{16}; the matrix corresponding to the *smallest* eigenvalue μ_{16} of $A_{(t)}$ (again, μ_{16} is generally negative). Each of the nine symmetric traceless 4×4 matrices U_i, and in particular U_{16}, has eigenvalues of the form $-\nu_{max} \leq -\nu_{min} \leq 0 \leq \nu_{min} \leq \nu_{max}$. Each pair $\langle \nu, -\nu \rangle$ corresponds to the two orientations in a "twisted pair" configuration (*i.e.* an orientation q and its 180-degree rotation around the translation t; *c.f.* [6, 13]), and result in the same algebraic error. Thus we may arbitrarily choose as initial estimate the eigenvector corresponding to the positive eigenvalue ν_{max}.

This approach to initializing the power method nearly always works well, but occasionally fails in neighborhoods of t in which the order of the smallest two (or more) eigenvalues of $\mathcal{A}_{(t)}$ changes—*i.e.* where the global minimum $q^* = \text{argmin}_q f(q,t)$ jumps between distant minima located in different basins when t is perturbed—as then the power method converges slowly, and possibly even to the wrong minimum (see Figure 1 and Figure 2). To work around this, we seed the power method also with the eigenvector corresponding to ν_{min}, since in this situation this vector must be nearly orthogonal to the suboptimal solution and hence closer to the true global optimum. We then choose as solution the best of the two eigenvectors found (if different). Table 2 lists the resulting algorithm, which makes use of S-HOPM, the more efficient and concise specialization of HOPM for symmetric tensors. In spite of the non-convexity of f that in theory could cause S-HOPM to fail [8, 9], we have found S-HOPM and HOPM to yield identical, globally optimal results (as verified exhaustively) in the several millions of cases we have tested.

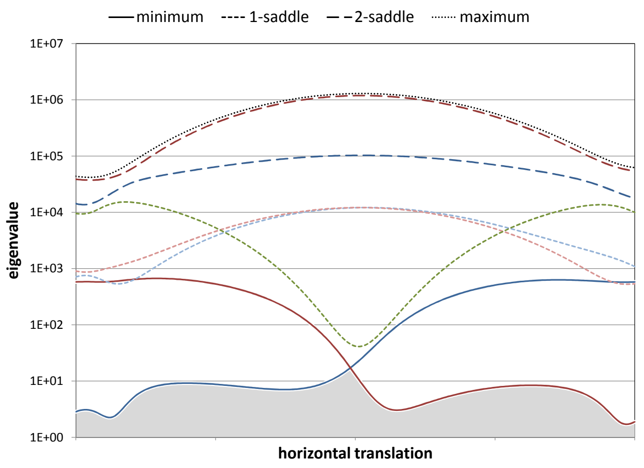


Figure 1. Tensor eigenvalues $\lambda_i(t)$, *i.e.* stationary values of $f(R, t)$, for the data in Figure 3 over a 1D slice of the translation domain (multiple eigenvalues have been consolidated). The minimal algebraic error $\min_R f(R, t)$ is the lower envelope of the functions $\lambda_i(t)$ (shaded area). To ensure convergence to the smallest eigenvalue near the "cusp," where the smallest two eigenvalues cross, the power method is seeded with two different starting points. (Note: The complete set of eigenvalues, shown here for illustrative purposes, was computed by exhaustive sampling and local descent, a procedure impractical for global optimization.)

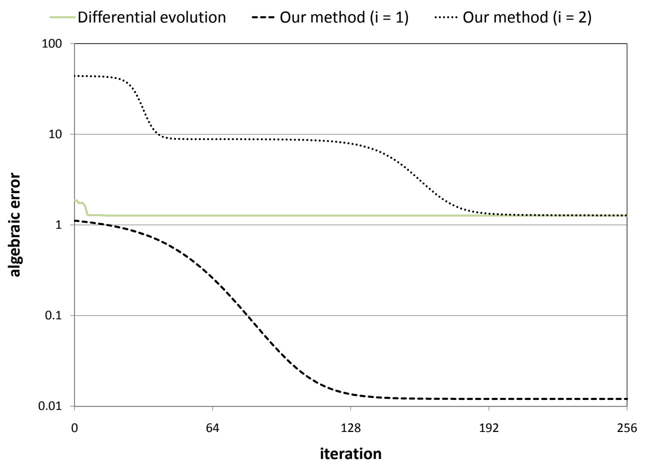


Figure 2. Convergence of the higher-order power method from two different initial estimates (*c.f.* Table 2). In this case, the $i = 2$ case converges to the same local minimum as differential evolution (DE), whereas $i = 1$ approaches the global minimum after roughly 128 iterations. Note that each power iteration is considerably cheaper to execute than one iteration of DE.

5. Unconstrained pose estimation

The two methods we have outlined above for determining camera extrinsics may be used in isolation in those situations where either the relative camera positions or orientations are known, or for which good estimates are available. When neither parameters are known, some bootstrapping mechanism is needed, as is also the case for most nonlinear pose estimators. To this end the normalized 8-point [4] and the 5-point method [10, 13, 21] are popular choices. Unfortunately, as is known and demonstrated below, for noisy correspondences or small baseline-to-scene-depth ratios, these methods may provide arbitrarily poor pose estimates.

When direct methods fail one has little recourse but to fall back on schemes like RANSAC to attempt to exclude outliers (in case of noisy data), or on sampling of the objective function (whenever noise is not the cause of failure) to identify an approximate initial pose. Due to the curse of dimensionality, uniformly sampling 5D pose space in as coarse as 15-degree increments requires on the order of one million evaluations of the algebraic error (Equation 9), which may be prohibitive in some applications. This suggest another utility of our method, which in effect reduces the 5D space to 2D: rather than spending $O(n^3)$ samples on orientation, our method finds for each sample translation the globally best orientation directly in $O(1)$ time. The dramatic reduction in the size of the search space more than offsets the higher cost of optimization versus blind sampling. In this sense, our method is a good complement to direct methods when those fail, and may seed further nonlinear optimization.

The idea of decomposing the 5D pose estimation problem into lower 2D or 3D subproblems is not new—only our approach to orientation optimization is. The direct solution to translation coupled with an outer nonlinear optimization of orientation, as proposed by Spetsakis and Aloimonos [20], and the converse direct (but suboptimal) method for estimating orientation from known epipoles by Hartley [5] are two examples. Both of these methods take ad hoc approaches to the outer optimization problem, however. Equipped with optimal solvers for both problems, we may instead alternate translation and orientation optimization, using the solution of one as input into the alternate optimizer, and taking arbitrarily large strides in pose space, possibly across "hill tops" in the objective function. Although each such step guarantees global optimality within a 2D or 3D subspace (a slice of pose space) and that the algebraic error is reduced, there are no guarantees that a global optimum over the full 5D pose space is attained. Nevertheless, using globally optimal subspace methods instead of local descent via gradient- or sampling-based methods is arguably a more reliable strategy. In practice, one would employ just a handful of iterations (we use 16 iterations as a limit) of alternate optimization over orthogonal subspaces to arrive at the correct basin containing the global optimum, from which one would then proceed to make strides in all five dimensions to avoid the need to "zig-zag" down a 5D valley (*c.f.* steepest descent versus conjugate gradients [16]).

6. Results

We now evaluate our method and compare it both with a nonlinear optimizer and with direct methods on imperfect, overconstrained inputs (> 8 points). Although each method evaluated may benefit from RANSAC, we chose not to employ it here in order to eliminate the influences of random chance and RANSAC parameter tweaking, and instead defer such a study to future work. In our evaluation we use one data set comprised of over one million points output by a dense correspondence code (Figure 3), as well as a collection of 100 point clouds of 100 random points each. The 3D points are distributed uniformly with a unit radial standard deviation, and are parameterized by their mean depth δ and standard deviation σ pixels of the Gaussian distributed random noise applied to their image projections. The cameras are for each data set positioned and oriented randomly, with a unit baseline and up to a 70 degree difference in orientation. All 3D points are visible to both cameras within 90-degree fields of view and 1024^2-pixel images.

We begin by examining how well orientation is optimized on the data in Figure 3. To gain some intuition for how our method performs, we plot in Figure 4 for each of 1025^2 unit translations (stereographically projected onto the unit square) the algebraic error for the best orientation found by three methods, including ours. *Differential evolution* (DE) [17]—a general-purpose sampling-based method—is used as a representative of *global* nonlinear optimizers, with a population size of 32 and up to 128 generations, *i.e.* as many as 4,096 rotation samples. Although more samples or multiple restarts could have been used, with these settings DE is already several times slower than our method. We also compare with Hartley's SVD method for known epipoles [5, Section 3(b)], slightly modified to compute essential rather than fundamental matrices.

Figure 4 shows that our method and DE produce the same globally optimal values $\min_R f(R,t)$ for most t, but in about 11% of cases DE gets stuck in a local minimum high above the global one, resulting in tall thin spikes snugly surrounded by tiny contours that appear as black dots. Meanwhile Hartley's method results in a vastly different, everywhere suboptimal landscape, and produces meaningful results only in the immediate vicinity of the globally optimal translation at the center. The difference in magnitude between local and global minima can better be appreciated in Figure 4(d). The intent of these plots is not to suggest that translation space be sampled uniformly as a basis for pose estimation, but simply to show that only our method reliably computes the optimal R regardless of t, which is important in alternating optimization when estimates of R and t are incrementally refined. We also point out that the failure rate of DE in 3D is likely to be amplified if run unconstrained in 5D pose space due to the combinatorial increase in local minima and samples needed.

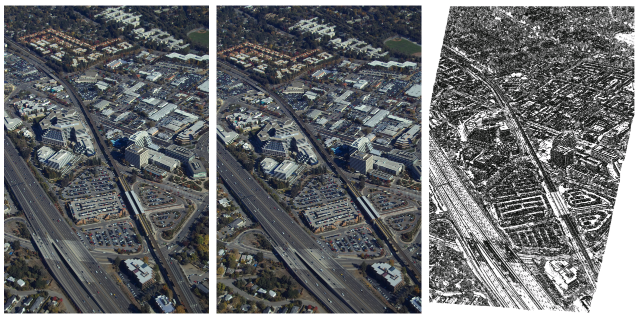


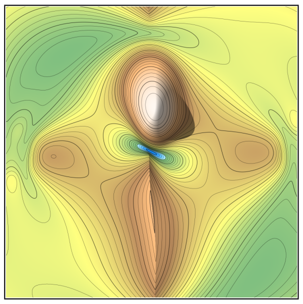
Figure 3. Image pair and 1.24 million approximate point matches.

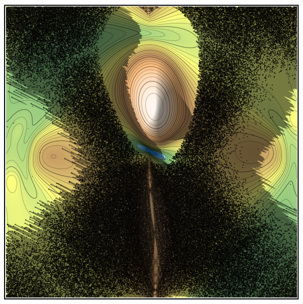
Figure 5 shows similar plots for one of the synthetic data sets. In (a) we also show how t changes between iterations of alternate optimization. Since this method monotonically reduces the error, it should not be surprising that the trajectories approximate the steepest descent lines of the 2D function $\min_R f(R,t)$. Note also the ridgeline separating two distinct basins of attraction. Although the alternating approach may take long strides, possibly even across this ridge, it does so rarely in practice.

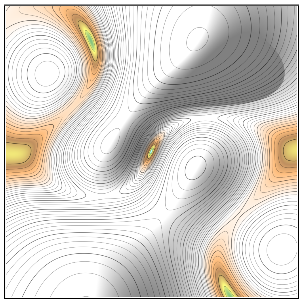
Oftentimes the relative camera positions are known to high accuracy, in which case it is desirable to constrain the optimization to orientations only. Figure 6 shows for the synthetic data orientation errors, measured as the angle of rotation separating two orientations, for our method, Hartley's, and the normalized 8-point [4] and 5-point [21] methods, which both ignore the translation constraint. We used Stewénius' Matlab implementation of the 5-point method [21], which benefited from our selection of the best orientation among the several hypotheses it generates. This figure indicates that our method consistently achieves lower error, sometimes by a wide margin (note the logarithmic scale). As observed by others [1], the 5-point method is often outperformed in the overconstrained case (> 5 points).

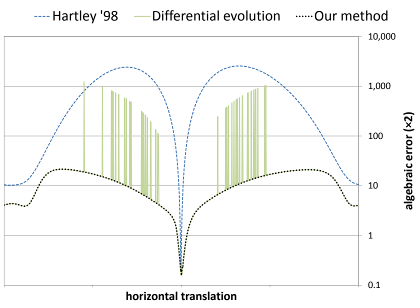
Figure 7 plots for unconstrained pose estimation the algebraic error ratio with respect to our alternating method, which was seeded with the pose found by the 8-point method. As expected, our rotation optimizer does worse than our unconstrained method on this measure, but only slightly so. Overall our alternating method performs best (unit error), except in (c). We here found that the combination of few points and relatively high noise and scene depth resulted in error functionals that favored "degenerate" poses discovered only by the 5-point method. Though low in algebraic error, these poses were often far from accurate. In fact, any significant difference in error between our two methods ought to signal a breakdown of the algebraic error functional, as occurs here on about 10% of the point clouds.

Due to its simplicity, our method is reasonably efficient. On the synthetic data the translation solver runs in about 200 μs; the rotation solver in 1–10 ms with a convergence criterion of six digits of precision, which is about 10 times


(a) Our globally optimal method.

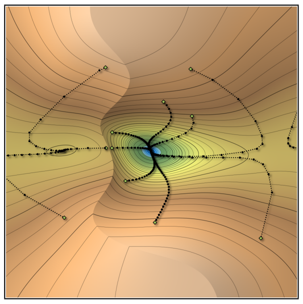

(b) DE (11% failure).

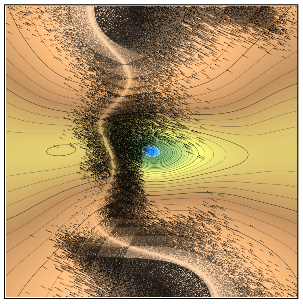

(c) Hartley '98 (100% failure).

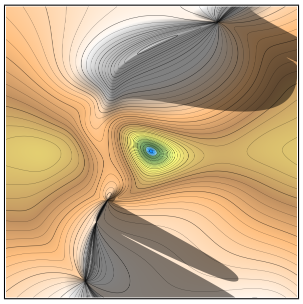


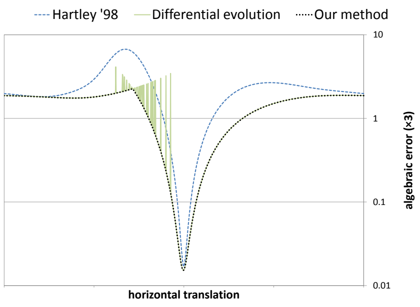
(d) Horizontal slice through center.

Figure 4. Contour plots (blue is low, white is high) with cast shadows of the function $\min_R f(R, t)$ plotted over the domain of all translations t. The 2D domain is a stereographic projection (flattening) of the unit sphere on which t lies; the function value plotted is the algebraic error for the corresponding best rotation R as computed by three methods: (a) Our method, which computes the true global minimum for any given t; (b) differential evolution [17], which often converges to a local minimum far above the true minimum, here manifested as isolated tall peaks; and (c) Hartley's linear method for fixed epipoles [5], which results in a function with vastly different topography except in a very small neighborhood around the globally optimal translation t (at the center of each plot; see (d)).


(a) Our globally optimal method.


(b) DE (8.4% failure).


(c) Hartley '98 (99% failure).



(d) Horizontal slice through center.

Figure 5. Comparison on synthetic data set comprised of one hundred random points (see also Figure 4). The ridgeline evident in (a) corresponds to the crossing of two eigenvalues $\lambda_1(t) = \lambda_2(t)$ (*c.f.* Figure 1), and consequently marks a thin band devoid of spikes in (b) where local and global minima are close in value. Seed points (green diamonds) and downhill trajectories in translation resulting from alternating R and t optimization are shown in (a). The illumination has been varied among these plots so as to highlight the topography.

slower than our implementation of [5]. By comparison, one rotation solve for the million-point data set takes 50 ms. We note that HOPM's rate of convergence depends on the distribution of eigenvalues—and in particular on the amount of noise—relative to the upper bound ρ, which in turn is proportional to the number of points. Hence our method works best in low-noise settings with moderately sized data. We anticipate that the rapidly growing interest in tensor eigenvalue methods will soon lead to more efficient solvers.

7. Conclusions

We have presented methods for computing the globally optimal relative pose from point correspondences for either fixed translation or orientation, which when used in tandem work to improve unconstrained pose estimates. We demonstrated improvements in pose accuracy over previous direct methods, whether translation or rotation parameters are constrained or not. Our methods rely on computing eigenvectors of symmetric tensors, and for orientation optimization we proposed using an iterative higher-order power method that has a straightforward implementation. Future work will investigate faster eigensolvers and improved seeding to increase the likelihood of simultaneously achieving global optimality in translation and orientation.

References

[1] M. Brückner, F. Bajramovic, and J. Denzler. Experimental evaluation of relative pose estimation algorithms. In *Third International Conference on Computer Vision Theory and Applications*, pages 431–438, 2008.

[2] K. C. Chang, K. Pearson, and T. Zhang. On eigenvalue problems of real symmetric tensors. *Journal of Mathematical Analysis and Applications*, 350:416–422, 2009.

[3] G. H. Golub and C. F. van Loan. *Matrix computations*. Johns Hopkins University Press, Baltimore, MD, 3rd edition, 1996.

[4] R. I. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593, 1997.

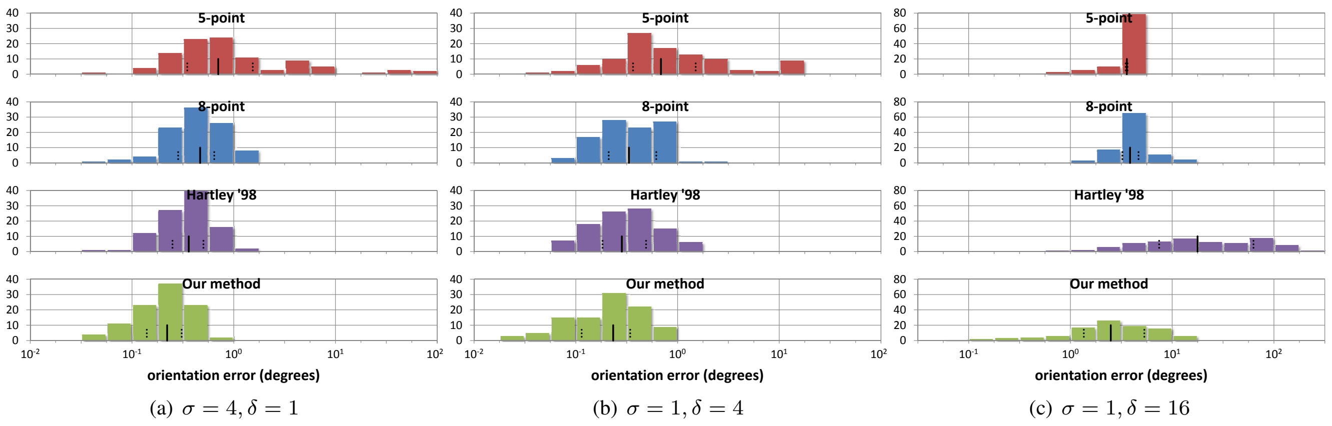


Figure 6. Orientation error distributions, quartiles (dotted), and median (solid) for noise level σ and scene depth δ. Our method and Hartley's [5] improve on the unconstrained 5- and 8-point methods by fixing the translation, except at large δ, where [5] performs poorly.

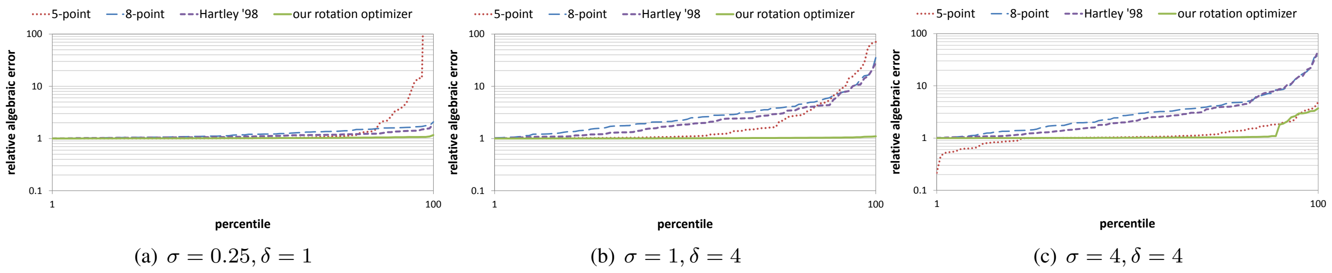


Figure 7. Algebraic error relative to our alternating method, sorted across 100 data sets at noise level σ and scene depth δ.

[5] R. I. Hartley. Minimizing algebraic error. *Phil. Trans. of The Royal Society*, 356(1740):1175–1192, 1998.

[6] R. I. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2004.

[7] U. Helmke. Essential matrix estimation using Gauss-Newton iterations on a manifold. *International Journal of Computer Vision*, 74(2):117–136, 2007.

[8] E. Kofidis and P. A. Regalia. On the best rank-1 approximation of higher-order supersymmetric tensors. *SIAM Journal on Matrix Analysis and Applications*, 23(3):863–884, 2002.

[9] L. D. Lathauwer, P. Comon, B. D. Moor, and J. Vandewalle. Higher-order power method—application in independent component analysis. In *International Symposium on Nonlinear Theory and its Applications*, pages 91–96, 1995.

[10] H. Li and R. Hartley. Five-point motion estimation made easy. In *18th International Conference on Pattern Recognition*, pages 630–633, 2006.

[11] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.

[12] M. I. A. Lourakis and A. A. Argyros. SBA: A software package for generic sparse bundle adjustment. *ACM Transactions on Mathematical Software*, 36(1), 2009. To appear.

[13] D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756–777, 2004.

[14] J. Philip. Critical point configurations of the 5-, 6-, 7-, and 8-point algorithms for relative orientation. Technical Report TRITA-MAT-1998-MA-13, Royal Institute of Technology, Feb. 1998.

[15] O. Pizarro, R. Eustice, and H. Singh. Relative pose estimation for instrumented, calibrated imaging platforms. In *VIIth Digital Image Computing: Techniques and Applications*, pages 601–612, 2003.

[16] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical recipes in C*. Cambridge University Press, 2nd edition, 1992.

[17] K. V. Price. *Differential evolution*. Springer, 1st edition, 2005.

[18] L. Qi. Eigenvalues of a real supersymmetric tensor. *Journal of Symbolic Computation*, 40(6):1302–1324, 2005.

[19] L. Qi, F. Wang, and Y. Wang. Z-eigenvalue methods for a global polynomial optimization problem. *Mathematical Programming*, 2007.

[20] M. E. Spetsakis and Y. Aloimonos. Optimal visual motion estimation: A note. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(9):959–964, 1992.

[21] H. Stewénius, C. Engels, and D. Nistér. Recent developments on direct relative orientation. *Journal of Photogrammetry and Remote Sensing*, 60(4):284–294, 2006.

[22] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment—a modern synthesis. In *Vision Algorithms: Theory and Practice*, pages 298–372, 1999.

[23] C. F. van Loan. The ubiquitous Kronecker product. *Journal of Computational and Applied Mathematics*, 123:85–100, 2000.